

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12]

Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12]

Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",^[28] but a brand that applies to the LLVM umbrella project.^[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14^[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014^[31] and was in post as of March 2024.^[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.^[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,^[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.^[3] As of November 2022 about 400 contributions had not been relicensed.^[35]^[36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.^[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.^[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.^[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,^[8]^[9]^[10] Common Lisp,^[11] PicoLisp, Crystal, CUDA, D,^[12] Delphi,^[13] Dylan, Forth,^[14] Fortran,^[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,^[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12]

Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12]

Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12]

Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12]

Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12]

Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12]

Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and

"inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12]

Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

LLVM, also called LLVM Core, is a target-independent optimizer and code generator.[5] It can be used to develop a frontend for any programming language and a backend for any instruction set architecture. LLVM is designed around a language-independent intermediate representation (IR) that serves as a portable, high-level assembly language that can be optimized with a variety of transformations over multiple passes.[6] The name LLVM originally stood for Low Level Virtual Machine. However, the project has since expanded, and the name is no longer an acronym but an orphan initialism.[7]

LLVM is written in C++ and is designed for compile-time, link-time, runtime, and "idle-time" optimization. Originally implemented for C and C++, the language-agnostic design of LLVM has since spawned a wide variety of frontends: languages with compilers that use LLVM (or which do not directly use LLVM but can generate compiled programs as LLVM IR) include ActionScript, Ada, C# for .NET,[8][9][10] Common Lisp,[11] PicoLisp, Crystal, CUDA, D,[12] Delphi,[13] Dylan, Forth,[14] Fortran,[15] FreeBASIC, Free Pascal, Halide, Haskell, Idris,[16] Jai (only for optimized release builds), Java bytecode, Julia, Kotlin, LabVIEW's G

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]

language,[17][18] Objective-C, OpenCL,[19] PostgreSQL's SQL and PLpgSQL,[20] Ruby,[21] Rust,[22] Scala,[23][24] Standard ML,[25] Swift, Xojo, and Zig.

History

The LLVM project started in 2000 at the University of Illinois at Urbana–Champaign, under the direction of Vikram Adve and Chris Lattner. LLVM was originally developed as a research infrastructure to investigate dynamic compilation techniques for static and dynamic programming languages. LLVM was released under the University of Illinois/NCSA Open Source License,[3] a permissive free software licence. In 2005, Apple Inc. hired Lattner and formed a team to work on the LLVM system for various uses within Apple's development systems.[26] LLVM has been an integral part of Apple's Xcode development tools for macOS and iOS since Xcode 4 in 2011.[27]

In 2006, Lattner started working on a new project named Clang. The combination of the Clang frontend and LLVM backend is named Clang/LLVM or simply Clang.

The name LLVM was originally an initialism for Low Level Virtual Machine. However, the LLVM project evolved into an umbrella project that has little relationship to what most current developers think of as a virtual machine. This made the initialism "confusing" and "inappropriate", and since 2011 LLVM is "officially no longer an acronym",[28] but a brand that applies to the LLVM umbrella project.[29] The project encompasses the LLVM intermediate representation (IR), the LLVM debugger, the LLVM implementation of the C++ Standard Library (with full support of C++11 and C++14[30]), etc. LLVM is administered by the LLVM Foundation. Compiler engineer Tanya Lattner became its president in 2014[31] and was in post as of March 2024.[32]

"For designing and implementing LLVM", the Association for Computing Machinery presented Vikram Adve, Chris Lattner, and Evan Cheng with the 2012 ACM Software System Award.[33]

The project was originally available under the UIUC license. After v9.0.0 released in 2019,[34] LLVM relicensed to the Apache License 2.0 with LLVM Exceptions.[3] As of November 2022 about 400 contributions had not been relicensed.[35][36]